



**UNIVERSITY OF ALBERTA**  
**FACULTY OF SCIENCE**  
Department of Computing Science

# **SOFTWARE PRODUCT MANAGEMENT**

**STANLEY'S PROBLEM: PART 2**  
**PRODUCT BACKLOG**

<b>Building a Product Backlog</b>	<b>1</b>
Bring in Trello	1
The MoSCoW Rule	2
Stanley's Must-Have Stories	3
Stanley's Should-Have Stories	3
Stanley's Could-Have Stories	4
Stanley's Won't-Have Stories	4
Optional Exercise	5

## Building a Product Backlog

The initial user stories you've written with your team and the client are gathered together and placed into the backlog with no particular order. At this point, your product backlog is an unordered lump of all the desired features that would ideally be built into the end product. Those stories are assigned with a unique user story identifier so that everyone has a frame of reference within the project.

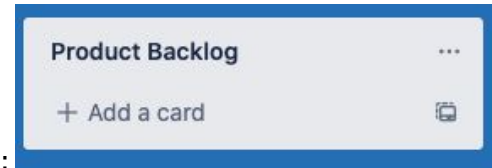
### Bring in Trello

Over the past few days, you start to realize that sharing files with a team that works remotely has been very challenging and you've decided to use a task management software called Trello.

You and your team study some Trello basics, learn how to visualize the Trello board's workflow, and how to track your work progress: If you are enrolled in **Course 2: Software Processes and Agile Practices**, please check out [this tutorial](#); alternatively, you can also read this [Google Doc version](#).

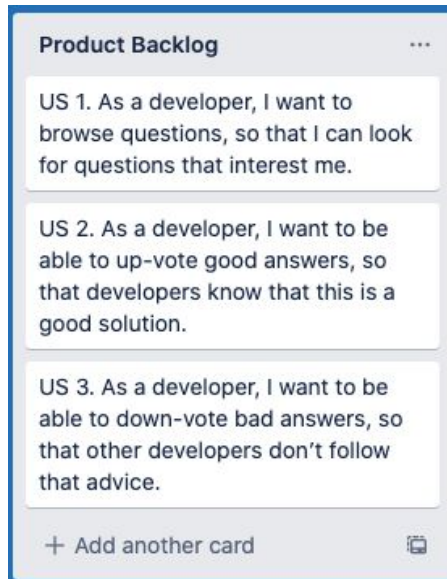
For this project, you start to organize your user stories in a Trello board. Everyone is excited to give it a try.





To start with, you create a list called “Product Backlog”:

Then start adding user stories to the cards:



You’ve added all your team members and the client, so everyone can see the [Trello board](#).

## The MoSCoW Rule

Once you have your unsorted backlog, the next thing you do is work with your client to give priority to each story so that the backlog is sorted from highest to lowest priority. The most high-value features are listed at the top of the backlog, and the least important ones at the bottom.

Stanley, the client, agrees to sort the stories for maximizing the value delivered to his company. However, he is not entirely sure how to do that since some features are equally important. To help with this, you decide to use the MoSCoW rule, and suggest that Stanley sort these user stories into four groups:

- Must-have: mandatory
- Should-have: mandatory if no time limits
- Could-have: could leave to next release if time runs out



- Won't-have: desired features but acknowledged they're not for this release

This helps Stanley figure out what he wants and soon he comes back with a list of stories for each group:

### Stanley's Must-Have Stories

US 9	As a developer, I want to sign up for an account, so that I can start activities that only registered users can perform.
US 13	As an account holder, I want to post questions, so that I can receive answers from other developers.
US 14	As an account holder, I want to be able to respond to questions, so that I can assist other developers.
US 1	As a developer, I want to be able to browse questions, so that I can look for questions that interest me.
US 6	As a developer, I want to be able perform a simple search for questions that match keywords in both the title and body fields, so that I can find related questions.
US 10	As a developer, I want to sign in with an email and password, so that I can start activities that only registered users can perform.
US 19	As an account holder, I want to be able to log out of my account, so that I can keep my account safe.

### Stanley's Should-Have Stories

US 7	As a developer, I want to be able to perform a full-text search of the question title, body, author, and category, so that I can find precisely what I want.
US 8	As a developer, I want to search for questions by category (eg. AI, Algorithms, Software Development etc.), so that I can quickly find what I'm looking for.
US 2	As a developer, I want to be able to up-vote good questions, so that developers know that this is a good solution.
US 3	As a developer, I want to be able to down-vote bad questions, so that developers don't follow this advice.
US 4	As a developer, I want answers to questions to be sorted by validity (# of up-votes), so that the best answers appear near the top.



US 15	As an account holder, I want to be able to comment on answers, so that I can elaborate on or ask questions about an answer.
US 16	As an account holder, I want to receive notification when someone answers my question, so that I can check a potential solution right away.
US 17	As an account holder, I want to bookmark a question that I find useful, so that I can navigate back to it when needed.
US 18	As an account holder, I want to put a question into a watchlist, so that I can get notifications when someone answers it.
US 21	As an account holder, I want my profile page to show the questions that I have asked or responded to so that others can see what I've posted.

### Stanley's Could-Have Stories

US 11	As a developer, I want to sign in with my Google account, so that I don't have to remember yet another username and password.
US 12	As a developer, I want to sign in with my Github account, so that I don't have to remember yet another username and password.
US 5	As a developer, I want to see a list of similar questions to the one I am viewing, so that I can find a solution faster.
US 20	As an account holder, I want to update my profile picture, so that people can see what I look like.

### Stanley's Won't-Have Stories

Stanley acknowledges that the invitation system is optional and is not for this release:

US 22	As an account holder, I want to invite another user to answer one question, so that a question can be solved faster.
US 23	As an account holder, I want to receive notification when someone invites me to answer a question, so that I can respond to it faster.

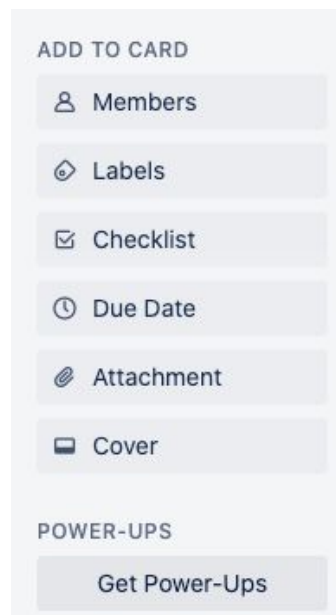
Based on this, the team has updated the [Trello board](#) to match with it.



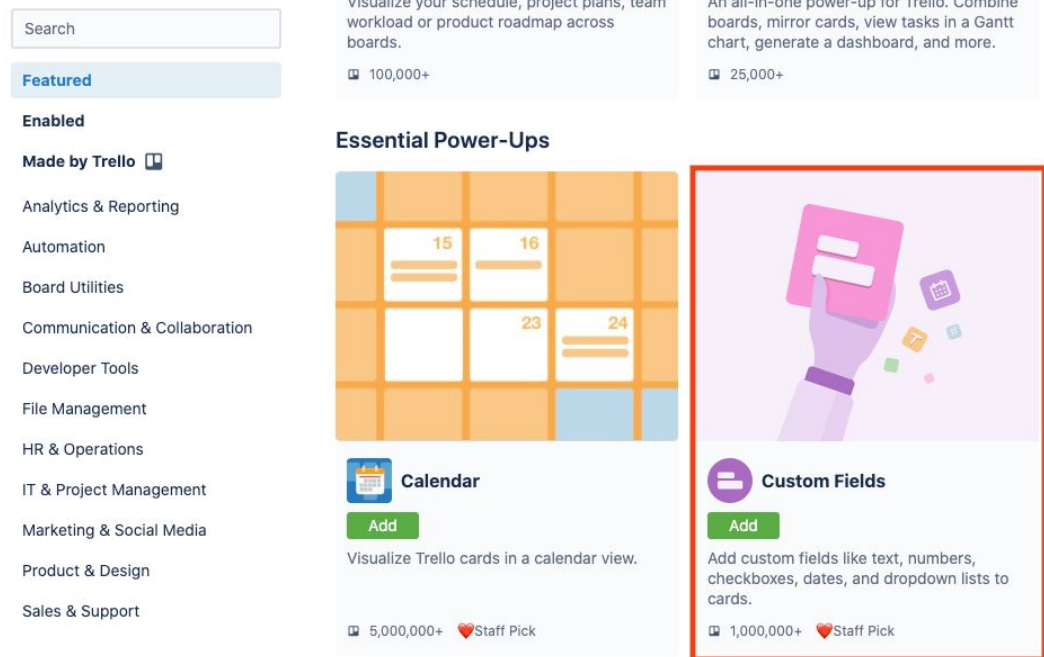
## Optional Exercise

Based on Stanley's preference, it's time to update the Trello board to match with prioritized User Stories. To complete this exercise, you will need to:

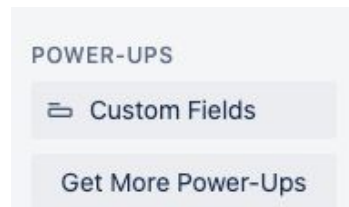
- Step 1. Create a Trello Board
- Step 2. Create a list called "Product Backlog"
- Step 3. Recreate all the cards from User Stories. At the end of this step, your board will look similar to [this board](#).
- Step 4. Assign Priority using Trello Power-ups: "Custom Field":
  - Step 4.1. Click on your card. On the right hand side menu, under "POWER-UPS", click on Get Power-Ups:



- Step 4.2. In "Featured", look for "Custom Fields". Alternatively, you can also search for it. Click on the great button "Add".



Now go back to our board, you can see that "Custom Fields" is added under POWER-UPS:



**Step 4.3** Click on Custom Fields, and select "New Field". Choose "Dropdown" as its type:

New Field

- Checkbox
- Date
- ✓ Dropdown
- Number
- Text

Add item...

Show field on front of card ✓

Create

Let's put "Priority" as its name, and start filling in items as options:

Must-Have; Should-Have; Could-Have; Won't-Have. (Note: press key "Enter" after entering every item)

**Step 4.4.** You can also assign some colors to each option.

Note that there's no particular color associated with any priority; alternatively, you can also use color for Risk, or even Estimate.

In this tutorial, we are adding colors for better illustration. Let's use red for must-have, yellow for should-have, and green for could-have. Leave "won't-have" with the default grey.

Options

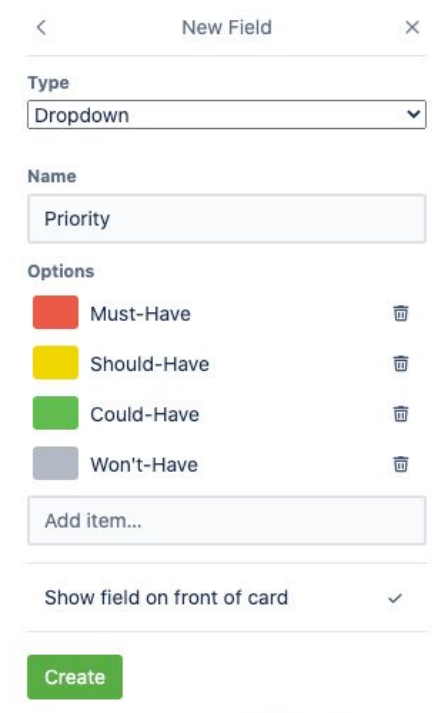
- Must-Have
- Should-Have
- Could-Have
- Won't-Have

Select a color

Color swatches: green, yellow, orange, red, purple, blue, cyan, light green, pink, dark grey

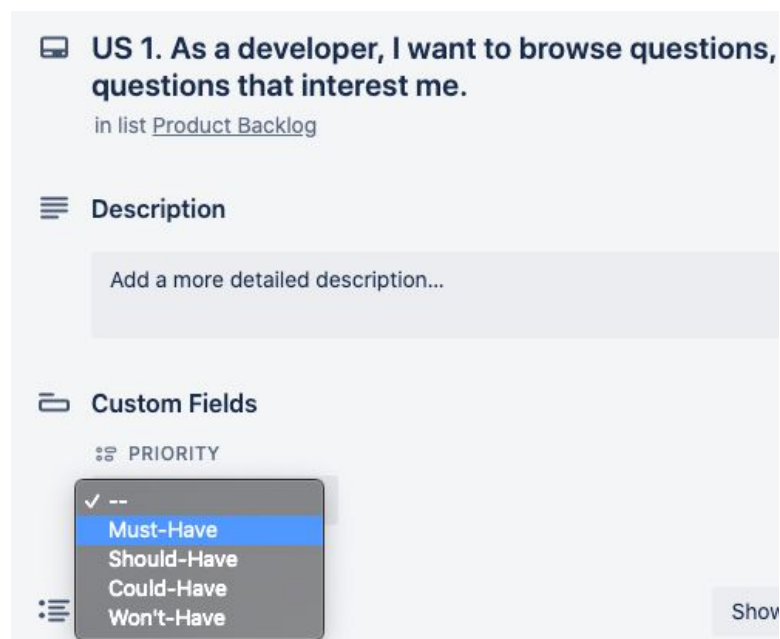


**In the end, it should look like this:**



Click **"Create"**

**Step 4.5.** In your card, you should be able to see Custom Fields. Since the first User Story is a "Must-Have" Story, let's assign a priority for it:



And in your **Product Backlog** list, your card should update automatically:



**Step 4.6.** Now it's your turn to:

- 1) Assign the correct priority for each User Story
- 2) Sort them in a correct order
- 3) And Compare your work with this [Trello Board](#).